

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
04.06.2003 Bulletin 2003/23

(51) Int Cl.7: **G06T 17/00**

(21) Application number: **01460075.3**

(22) Date of filing: **30.11.2001**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU
MC NL PT SE TR
 Designated Extension States:
AL LT LV MK RO SI

• **Gioia, Patrick**
35510 Cesson-Sevigne (FR)

(71) Applicant: **FRANCE TELECOM**
75015 Paris (FR)

(74) Representative: **Vidon, Patrice**
Cabinet Patrice Vidon
Le Nobel (Bât. A)
Technopôle Atalante
2, allée Antoine Becquerel
BP 90333
35703 Rennes Cedex 7 (FR)

(72) Inventors:
 • **Cotarmanac'h, Alexandre**
35000 Rennes (FR)

(54) **Adaptative method for displaying a graphic scene on a terminal based on wavelets coding**

(57) The invention regards a method for rendering a graphic scene on a terminal, said graphic scene implementing a hierarchical tree comprising at least one node associated to an object of said scene.

According to the invention, at least one of said objects is defined by a wavelet-coded mesh, as a base mesh and wavelet coefficients corresponding to local modifications of said base mesh.

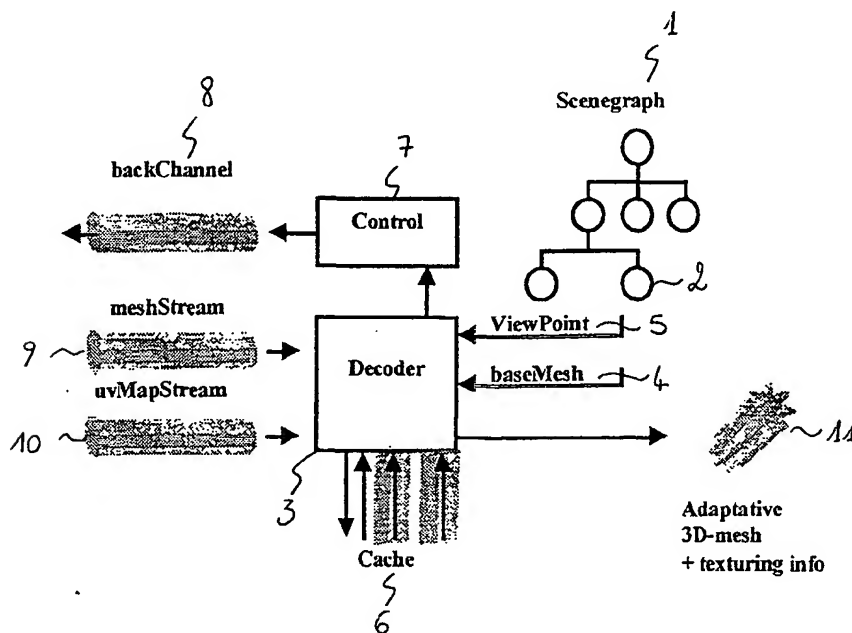


FIGURE 1

Description

[0001] The invention regards the rendering and displaying of an audio-visual scene on an user terminal. More precisely, the invention regards the representation of a multimedia graphic scene, comprising several objects interacting with one another, on a terminal, and especially, but not exclusively, graphic scenes coded according to the MPEG4 standard ("Motion Picture Expert Group").

[0002] By object, it is meant, here and in the whole document, a representation of a natural or synthetic object that has an audio and/or visual manifestation. (The reader can refer to annex 1 for definitions of some of the most important words used in the present application. The content of all the annexes to the present document are also part of the present application.)

1. Prior art

[0003] Two main standards, namely MPEG-4 (BIFS) and VRML (annex 2 provides a list of all the abbreviations and symbols used in this document), describe some rules for multimedia representation. In both standards, the management of three-dimensional objects offers two mechanisms for allowing an adaptive representation of a multimedia scene, namely :

- a first mechanism called LOD, for "Level Of Details". According to this mechanism, a single object is associated to different geometrical representations (for example, a coarse representation, and several successive more detailed representations). According to the distance between the object and an observer, the terminal choses the best fitted representation;
- a second mechanism, called Mesh3D or 3DMC, which is a method for compressing meshes, allowing to add polygons to the current mesh representation. It is thus possible to modify the number of polygons used in the mesh representation, in order to adapt the representation to an allowed budget of polygons, which may be a function, for example, of the terminal capacity.

[0004] A drawback of the first mechanism is that a great number of representations is associated to each object, which leads to the appearance of visual artefacts when changing from a first representation to the next one. Such a phenomenon is also known as "popping".

[0005] Such a state of the art mechanism is hence memory consuming, and lacks flexibility : each object is actually associated to a limited number of representations, and there is no criterion, apart from the distance from a virtual observer to the object, to decide which representation of the object should be displayed.

[0006] An advantage of the Mesh3D mechanism is that it offers a high compression ratio, and allows for an incremental building of the 3D model. This mechanism is based on Topological Surgery and Progressive Forest Splits. The principle is to encode topologically, and in a predictive way, some groups of connected local simplifications (vertex splits). The method is intended to provide both compression and progressiveness. This last feature is very important since it permits partial transmission and/or reconstruction according to bit rate or terminal rendering capabilities.

[0007] However, this method fails to provide both state of the art compression and decoding convenience.

[0008] First, Mesh3D is progressive but not hierarchical. The details can be added locally, but there is no obvious way to relate the coarsest versions of the mesh and further details added over it. Thus it is not well suited for animation, as hierarchical methods can be. Moreover, on-demand transmission of refinements and view-dependent reconstruction are quite difficult, since details are not well-localised and cannot be added independently. Actually, a drawback of this mechanism is that there is no visual quality criterion which can help decide what is the best number of polygons to be used in the mesh, depending on the viewpoint of an observer. It is thus very difficult, according to this mechanism, to simplify a mesh, by suppressing polygons which are not visually relevant for an observer.

[0009] In other words, such a mechanism allows an incremental, but non-adaptive, transmission of data.

[0010] Another main drawback of using 3DMC with recent 3D scenes is the compression ratio, which remains too low to enable fast reconstruction in the context of 3D model animation.

[0011] Still another drawback of 3DMC is that a 3DMC decoder is quite complex.

2. Presentation of the invention

2.1 Goal

[0012] It is an object of the invention to overcome these different drawbacks of the prior art.

[0013] More specifically, the invention aims at providing a method for allowing a hierarchical, efficiently compressed representation, enabling fast and selective reconstruction, in the context of 3D model animation.

[0014] Another object of the invention is to provide such a method which is well suited for network environments.

[0015] The invention also aims at providing a method for rendering a graphic scene on a terminal, which is well suited for animation of objects within the scene.

5 [0016] Another object of the invention is to provide such a method which allows on-demand transmission of refinements and view-dependent reconstruction of objects.

[0017] The invention also aims at providing such a method, which allows to render graphic scenes presenting an improved granularity compared to the methods of the art.

[0018] Still another object of the invention is to provide such a method allowing to introduce new functionalities into graphic scenes, and in particular adaptive transmission and decoding.

10

2.2 Main features of the invention

[0019] These objects as well as others which will appear subsequently, are achieved by means of a method for rendering a graphic scene on a terminal, said graphic scene implementing a hierarchical tree comprising at least one node associated to an object of said scene.

15

[0020] According to the invention, at least one of said objects is defined by a wavelet-coded mesh, as a base mesh and wavelet coefficients corresponding to local modifications of said base mesh.

[0021] Advantageously, said graphic scene is of the MPEG4 type.

20

[0022] Preferentially, for each of said objects defined by a wavelet-coded mesh, said corresponding node is associated, within a node interface of said hierarchical tree, to :

- a field relating to said base mesh, called BaseMesh ;
- a field relating to at least one rendering stream of said object, comprising said wavelets coefficients, called Mesh-Stream,

25

and said BaseMesh and MeshStream fields can be transmitted and/or processed independently.

[0023] According to an advantageous feature of the invention, said terminal has a back channel, allowing to transmit at least one piece of information relating to said terminal and/or to an user of said terminal to a graphic scene server.

30

[0024] Preferentially, for each of said objects defined by a wavelet-coded mesh, said corresponding node interface comprises a field allowing to indicate said back channel.

[0025] Advantageously, said piece of information belongs to the group comprising

35

- a piece of information relating to the position of said user in said graphic scene, called position information ;
- a piece of information relating to a direction of observation of said user, called direction information ;
- a piece of information relating to the state of a cache of said terminal.

[0026] According to a preferred embodiment of the invention, said node interface also implements at least one of the fields belonging to the group comprising :

40

- a field relating to the texture of said object;
- a field indicating a frequency at which said terminal transmits at least one state information to a server ;
- a field relating to a field of view of said user, called FOV ;
- a field indicating a quality of rendering of said object, called Quality.

45

[0027] Advantageously, said node interface also comprises a field, called FocusPoint, allowing to indicate at least one area of said object for which an user wishes to get a quality of rendering greater than a predetermined threshold.

[0028] Preferentially, said Quality field pilots a relevance threshold for said wavelets coefficients, the wavelets coefficients lower than said relevance threshold not being applied to said mesh.

50

[0029] According to an advantageous feature of the invention, at least one of said objects may be animated by modifying in time, within said BaseMesh field, coordinates of at least one vertex of said base mesh, the wavelets coefficients of said MeshStream field remaining unchanged.

[0030] Advantageously, a step of editing said mesh allows to modify an existing object by acting on at least one of the elements belonging to the group comprising

55

- said base mesh;
- said wavelets coefficients ;
- a current mesh associated to said existing object within said rendered scene.

Preferentially, said rendering stream of said MeshStream field is organised into packets of at least one wavelets

coefficient.

Advantageously, said packets of said rendering stream are of at least two types of packets, among the types belonging to the group comprising

- base packets, comprising data which are essential to the rendering of said mesh defining said object;
- 5 - refinement packets ;
- bitplanes refinement packets, said refinement packets and said bitplanes refinement packets comprising data for refining said mesh.

[0031] Preferentially, said refinement packets comprise at least one wavelets coefficient allowing to refine the rendering of at least one area of said object.

[0032] Advantageously, said bitplanes refinement packets allow to refine the preciseness of at least one wavelets coefficient.

[0033] According to a first embodiment of the invention, said packets are transmitted from a server to at least one terminal within a single rendering stream.

15 [0034] According to a second embodiment of the invention, said packets are transmitted from a server to at least one terminal within at least two parallel data streams, corresponding respectively to a base stream, comprising at least one base packet, and to at least one refinement stream, comprising at least one refinement packet and/or at least one bitplanes refinement packet.

[0035] Preferentially, said information relating to the state of a cache of said terminal indicates at least:

- 20 - a number of wavelets coefficients associated to each packet stored within said cache ;
- a number of bitplanes associated to each packet stored within said cache.

[0036] Advantageously, said wavelets coefficients are organised into at least two separate subsets, extracted from a global data stream allowing to rebuild said object, each of said subsets corresponding to areas of said object, and undergoing independent coding, localisation data, which allow to locate said separate subsets within said global data stream, being inserted within said global data stream, so as to allow to distribute said wavelets coefficients among said packets, in view of a selective transmission of at least one of said subsets towards said terminal.

[0037] According to a preferred embodiment of the invention, said subsets undergo a coding of "zero-tree" type.

30 [0038] According to an advantageous feature of the invention, each type of packet is structured into subpacket(s) belonging to the group comprising

- for a base packet :
 - 35 - a subpacket relating to scale coefficients, allowing to modify the position of at least one vertex of said base mesh;
 - at least one subpacket relating to wavelets coefficients associated to a zero-tree, containing consecutive bitplanes and at least the first one ;
- 40 - for a refinement packet:
 - at least one subpacket relating to wavelets coefficients associated to a zero-tree, containing consecutive bitplanes and at least the first one ;
- 45 - for a bitplanes refinement packet:
 - at least one subpacket relating to wavelets coefficients associated to a zero-tree, containing consecutive bitplanes.

50 [0039] Preferentially, said global data stream comprises a header, comprising at least some of said localisation data, and a wavelets coefficients area, comprising a subarea identified by said localisation data for each of said subsets.

[0040] Advantageously, said header also comprises at least some of the information belonging to the group comprising

- 55 - the number of faces of said base mesh ;
- the type of wavelets ;
- information relating to said object;
- information relating to the coding of said localisation data.

[0041] According to an advantageous feature of the invention, a wavelets coded texture is also associated to at least one object defined by a wavelets coded mesh.

[0042] The invention also concerns a terminal rendering at least one graphic scene according to the method described hereabove, as well as a server of graphic scenes transmitting to at least one client terminal a graphic scene designed to be rendered according to the previously described method.

[0043] The invention also relates to a data storage medium comprising at least one graphic scene designed to be rendered by at least one client terminal according to the method described hereabove.

[0044] The invention covers as well a method for coding a graphic scene implementing a hierarchical tree comprising at least one node associated to an object, wherein at least one of said objects is defined by a wavelets coded mesh, as a base mesh and wavelets coefficients corresponding to local modifications of said base mesh. According to the invention, said graphic scene is designed to be rendered on at least one client terminal according to the previously described rendering method.

[0045] The invention still relates to a graphic scene implementing a hierarchical tree comprising at least one node associated to an object, wherein at least one of said objects is defined by a wavelets coded mesh, as a base mesh and wavelets coefficients corresponding to local modifications of said base mesh, as well as a node of a hierarchical tree representative of a graphic scene, associated to an object defined by a wavelets coded mesh, as a base mesh and wavelets coefficients corresponding to local modifications of said base mesh.

[0046] The invention also relates to a signal to be transmitted to at least one terminal for rendering a graphic scene on said terminal, said signal comprising data representative of said graphic scene implementing a hierarchical tree comprising at least one node associated to an object of said scene.

[0047] According to the invention, said signal comprises at least one object defined by a wavelets coded mesh, as a base mesh and wavelets coefficients corresponding to local modifications of said base mesh.

[0048] All the features described hereabove in relation to the method for rendering a graphic scene of course also apply to the signal according to the invention, although they are not explicitly stated here, for conciseness purpose.

[0049] The invention also relates to a device comprising means for emitting a signal as previously described, as well as a device comprising means for receiving a signal as previously described.

2.3 Drawings

[0050] Other features and advantages of the invention will be more clearly apparent upon reading the following description of one preferred embodiment of the invention, given as an illustrative and non-limiting example and the appended drawings wherein :

- Figure 1 discloses the structure and the functioning of a decoder according to the invention, allowing to render a graphic scene on a terminal;
- Figures 2a to 2d disclose the structure of the packets comprised in the signal according to the invention and decoded by the decoder described in figure 1 ;
- Figure 3 illustrates an embodiment of the decoding process according to the invention, allowing to provide a reconstructed 3D wavelet mesh from a base mesh and a 3D wavelet mesh bitstream;
- Figure 4 illustrates the notion of offspring of a coefficient.

[0051] Before describing in details a preferred embodiment of the invention, we recall briefly the main features of the two main coding techniques of objects or images, namely :

- the classical method based on the subdivision of the faces of a polygonal mesh;
- the wavelet coding method.

[0052] Second generation wavelets have proven to be a powerful tool for scalable representation of 3D surfaces.

[0053] One important property of wavelets is that they are well localised in space, which enables efficient selection of significant wavelets according to the position of the refinements they induce, as well as their frequency or magnitude. This selection can of course be applied at the reconstruction stage, but in a client/server configuration, the server can also use this information to transmit only the necessary refinements to the terminal.

[0054] In the classical subdivision surfaces setting, objects are described by a coarse mesh (called the *base mesh*) along with a subdivision rule that represents an *almost-everywhere* smoothing : the subdivision rule is iterated so that the resulting sequence of meshes converge to the initial surface. Hence, given this rule, no extra information has to be transmitted to get the smooth object.

[0055] In a nutshell, the principle of geometrical wavelets is to locally weight the application of a given subdivision rule over the base mesh. Information is thus added at various stage of the subdivision rule iteration, as described in

ISO 639-2 :1998, Code for the representation of names of languages - Part 2 : Alpha-3 codes. Hence a subdivision surface can be regarded as a wavelet representation with all coefficients set to 0.

[0056] It is important to note that the initial models (those to be encoded) suitable for each of these two techniques are from two basically different kinds : subdivision surfaces are adapted to models with large regular areas, while wavelets are useful for objects with many chaotic "high frequency" peaks. For example, representing a noised object with subdivision surfaces will require too complex a base mesh, which has to contain high frequencies.

[0057] Though wavelet representation performs high decorrelation of the surfaces geometry, their encoding can lead to further considerable gain in compression. The state-of-the-art techniques for efficiently encoding both classical or geometrical wavelets are zero-trees. Zero-trees provide progressive encoding of the *significance maps*, i.e. parts of the object which are pertinent for a given threshold. The objective is to mark the largest amount of non-pertinent coefficients with minimum information.

[0058] Zero-trees have initially been developed in the context of 2D image coding (as described for example by M. Lounsbery et al. "Multiresolution Analysis for Surfaces of Arbitrary Topological Type", ACM Trans. On Graphics, 16(1), Jan. 1997 or by A. Khodakovsky et al. in "Progressive Geometry Compression", SIGGRAPH 2000 proceedings), and extended to 3D shapes (as described for example by A. Said and W. A. Pearlman in "Image compression using the spatial-orientation tree" In. Proc. IEEE Int. Symp. Circuits and Syst., Chicago, IL, pages 279-282, May 1993). The original algorithms can be directly applied provided that a hierarchy is expressed between coefficients, as it is canonically the case in the 2D setting. One drawback of the impressive compression ratios obtained by zero-trees is that, up to date, the order of decoding was imposed.

[0059] Suppose now that a server transmits geometric data to a terminal with unknown and/or limited processing power. As the portion of a 3D scene actually viewed is generally a small part of the whole scene, most part of the corresponding bitstream may not be needed by the client.

[0060] With the traditional zero-tree embedded representation, the client can start navigating in the 3D scene just after the reception of the base mesh, which is then refined during the walkthrough. But as the order of arrival of the coefficients is imposed by the zero-tree representation, the terminal may have to load many useless wavelet coefficients before the ones it actually needed.

[0061] The inventors have had the idea that a great improvement could be achieved if the terminal could be able to ask the server some parts of the bitstream representation of the wavelet coefficients, and if the server could be able to select the relevant parts of this bitstream from information about the client (e.g. view direction, position, viewed facets, ...) provided through the back-channel. The main problem is that enabling the change in the order of arrival of the coefficients would imply non-negligible loss in term of compression.

[0062] As will be described further in more details, the inventors have found an inventive solution to this problem, by building a bitstream, composed of isolated zero-trees on areas independent enough to have comparable compression ratios, but small enough compared to the whole model to let the corresponding groups of wavelet coefficients have local influence on the base mesh.

[0063] We now define the principles of adaptive 3D-mesh decoding according to the invention.

[0064] For clarity and simplicity purposes, we consider, from now on, a MPEG4-type scene, comprising several objects, which may be either 2D or 3D objects.

A.1.1 Scope

A.1.1.1 Architecture

[0065] Figure 1 describes a simplified decoder block-diagram. Such a decoder is for example comprised within the user terminal designed to display the considered graphic scene.

[0066] Such a graphic scene is associated to a scene graph 1 implemented as a hierarchical tree comprising a plurality of nodes. Node 2 is for example associated to an object defined by a mesh (or a texture) coded according to the wavelets encoding method.

[0067] The decoder 3 decodes the base mesh 4 associated to node 2, so that the corresponding object can be rendered by the terminal at its coarsest definition.

[0068] Decoder 3 also exchanges information with a cache 6 associated to the terminal : it can ask the cache 6 about its state (for example, it can ask for the list of wavelet coefficients associated to node 2 stored in cache 6, or for the number of bitplanes associated to each packet store in cache 6), and it can receive some data necessary for displaying the considered object from the cache 6 (for example, the cache 6 sends to decoder 3 the wavelet coefficients necessary for refining a portion of the base mesh 4, as a function of the view point 5 of a virtual observer).

[0069] Decoder 3 also provides the control entity 7 with information relating to the terminal and/or relating to the user, such as some information about the view point 5 of an user of the terminal, the direction of observation of the user, or the state of cache 6 for example.

[0070] The control entity 7 sends these information to a server, through the back channel 8.

[0071] The server (not illustrated on figure 1) computes the wavelet coefficients which are necessary to the terminal for displaying the scene, as a function of the information received through the back channel 8, and sends them to decoder 3, through the MeshStream entity 9. In other embodiment of the invention, these wavelet coefficients may have been pre-computed. The server can also computes some data needed by the terminal to render the texture of the object considered, and sends them through uvMapStream 10.

[0072] The decoder 3 decodes the data received through MeshStream 9, through uvMapStream 10, and from the cache 6, so as to build a representation 11 of the object, from a base mesh and from wavelets coefficients corresponding to local refinements of the mesh.

A.1.1.2 Walkthrough.

[0073] We now briefly describe the walkthrough in relation to figure 1.

- First the BIFS Node 2 is decoded.
- The BaseMesh 4 is decoded and instantiated.
- The texture is decoded and instantiated.
- Now, it's possible to render the node 2 at its coarsest definition.
- When the BaseMesh 4 has been instantiated, the following streams, if they exist, are set up :

⇒ MeshStream 9
 ⇒ BackChannel 8
 ⇒ UvMap 10

- Data from MeshStream 9 are received, they are cached depending on the viewpoint and quality fields.
- Depending on viewpoint and quality fields, the adaptive mesh is built using the BaseMesh 4, its TextureCoordinate and available coefficients that have been gathered both in MeshStream 9 and UvMap 10.
- Depending on the frequency field and on the viewpoint, data is sent through the BackChannel 8. This contains information about viewpoint and cache state.
- Depending on the viewpoint and on the cache state, data is removed from the cache 6.
- When the coordinates of the BaseMesh 4 are animated, modified, interpolated, the adaptive mesh is re-built.

A.1.2 Node specification

A.1.2.1 WaveMesh

A.1.2.1.1 Node interface

[0074]

```
WaveMesh {
    field      MFString    backChannel    []
    field      SFNode      baseMesh       NULL
    field      MFString    meshStream     []
    exposedField SFNode     texture       NULL
    exposedField SFFloat    frequency     1.0
    exposedField SFFloat    FOV           45
    exposedField SFInt32    quality       1
}
```

[0075] The binary encoding of this node is described in more details in annex 3.

A.1.2.1.2 Functionality and semantics

- 5 [0076] The **WaveMesh** node is used to represent adaptively multi-resolution polygonal models. It allows representing at any moment a 3D-model at a resolution which depends on the distance of the object with respect to the viewpoint. Transmission of such a model consists of two parts : a base mesh which describes the model at the coarsest level of detail, and series of refinements (wavelet coefficients) which can be carried on one or more elementary streams (one can refer to annex 1 of this document for a definition of the term "bitstream").
- 10 [0077] The **BackChannel** field specifies, if present, an upstream channel which allows a server to monitor which series of refinement are needed at the terminal. Thus in an interactive scenario (client-server interaction), refinements are sent according to the client state ((one can refer to annex 1 of this document for a definition of the term "back-channel bitstream").
- 15 [0078] The **BaseMesh** field shall specify either a Hierarchical3DMesh node or an **IndexedFaceSet** node. In the former case, the stream containing the mesh shall consist of a single access unit, and it will be explicitly instantiated as an **IndexedFaceSet**. In both cases, the structure of the (resulting) **IndexedFaceSet** will be preserved, namely deletion and insertion of new faces and/or vertices is explicitly forbidden. However, replacing the values of the coordinates is allowed.
- 20 [0079] As one will note, the inventors have chosen, in a preferred embodiment of the invention, to limit the functionalities inside the node, instead of redefining new nodes.
- [0080] It can also be noted that instantiating explicitly BaseMesh as an IndexedFaceSet allows animating the node.
- [0081] The **MeshStream** field shall specify the data source for wavelet coefficients, as will be explained further in more details. The terminal is responsible for maintaining an internal representation of the mesh to be rendered consisting of the BaseMesh to which apply the **MeshStream** data.
- 25 [0082] TextureCoordinate is provided by the IndexedFaceSet. The **TextureCoordinate** node shall apply on a per vertex basis. Texture coordinate is interpolated when the mesh is refined.
- [0083] The texture exposedField allows to map a texture to the geometry. Texturing rules are as of VRML, applied to the BaseMesh instantiated IndexedFaceSet
- [0084] The frequency exposedField gives a preferred frequency at which the terminal shall give its state by the means of the BackChannel stream if it exists.
- 30 [0085] As it can be noted, if the BackChannel does not exist this field has no effect.
- [0086] The **quality** exposedField allows to parametrize the quality of the rendered object. The range of values for this field is [0..2] where 0 stands for low quality, 1 for normal quality, 2 for best quality. Those values do not have a more explicit semantic, that is, it is a hint for the terminal.
- 35 [0087] For instance a quality value of 0, could mean using a very fast but approximate metric for choosing relevant wavelet coefficients. The player is responsible for adapting to such changes.
- [0088] The WaveMesh node is seen by the player as an **IndexedFaceSet** node container. That is to say, the player has access to the BaseMesh field.

40 A.1.3 MeshStream : the wavelet mesh coefficient stream

- [0089] The MeshStream is the stream conveying the wavelet coefficients which are necessary to represent adaptively a 3D-Mesh.
- 45 [0090] The different fields described hereabove as part of a node interface according to the invention allow to provide several interesting functionalities to an user of the terminal on which the graphic scene is rendered, namely :
- it allows to animate an object within the graphic scene, by animating the base mesh associated to this object. Actually, as described above, the BaseMesh field internally behaves as an IndexedFaceSet type mesh, on which it is possible to modify the coordinates of the vertices. Animating the vertices of the base mesh will thus allow to
- 50 animate the whole reconstructed mesh, at the given level of refinement or rendering considered. The wavelets coefficients will actually apply on the animated base mesh. Such an animation could be described as follows :

55


```

DEF CI CoordinateInterpolator
{
5      ..
      }
DEF WV WaveMesh
10 {
      baseMesh DEF IFS IndexedFaceSet
      {
15          coord DEF COORD Coordinate { }
      }
20 }
ROUTE CI.value_changed TO COORD.point

```

25 Another embodiment of such a functionality could also consist in adding a field called "points" to the node interface described hereabove.

- it also allows to refine some parts of the mesh, as requested by the user. This is achieved by adding a field called exposedField_SFVec3f_focus_point (also called Focus Point in this document), defined in the local coordinate system associated to the base mesh. For example, an user of the terminal can select a portion of the object he wishes to refine in more details, by clicking on such a portion with a mouse associated to the terminal. It is thus possible to route on the Focus Point field, in order to ask for a greater quality around the point selected by the user. It is also possible, for example, to ask the user about the size of the portion surrounding the selected point of the mesh he wishes to refine, which is somehow equivalent to setting a second FOV type field.
- such a node interface also allows to adjust the threshold above which the wavelet coefficients have to be taken into account. This is made possible through the Quality field, which has been previously described as being a discrete field which value could either be 0, 1 or 2, but which could also take continuous values in another embodiment of the invention. The wavelet coefficients are actually taken into account when their value is greater than a relevance threshold, which is a function of the distance from a virtual observer (that is to say, the user of the terminal) to the object. The Quality field could thus pilot the relevance threshold, so as to allow the user to decide with accuracy which wavelet coefficients should be taken into account for rendering the object.
- the structure of the node interface described above also allows to edit a mesh, thanks to the BaseMesh and FocusPoint fields, through which it is possible to modify the base mesh as well as the wavelet coefficients. Exposing the base mesh allows to animate the object, as described hereabove, but also allows to edit the mesh on the associated base mesh. Moreover, by adding the functionalities described above in relation to the Focus Point and Quality fields, it is possible to select a mesh at a given level of refinement, and to modify the associated wavelet coefficients, thus allowing an easy and multi-resolution editing. It is thus possible to modify an object by acting either on the base mesh, on the wavelet coefficients, or on the mesh associated to an object for a given level of rendering.

A.1.3.1 Stream Configuration : DecoderSpecificInfo

A.1.3.2 Syntax

```

5  [0091]

        class Wmesh_DecoderConfig extends DecoderSpecificInfo : bit(8)
10  tag=DecSpecificInfoTag {
            bit(1) Wtype;
            bit(5) NbBP;
15  double Cmax;
            bit(5) NbBpSC;
        }
20

```

A.1.3.3 Semantics

[0092] NbBP : the number of bitplanes. (typically 12 or 16)

25 Wtype : the type of wavelet.

0 : lifted midpoint subdivision with one vanishing moment.
1 : Loop

30 NbBpSC : the number of bit per scale coefficient.
Cmax : maximum wavelet/scale coefficient magnitude.

A.1.3.4 Bitstream syntax.

35 [0093] The MeshStream is the stream conveying the wavelet coefficients which are necessary to represent adaptively a 3D-Mesh.

[0094] Figures 2a to 2d respectively describe the file structure and the bitstream structure for every type of packets, namely the base packets, the refinement packets, and the bitplanes refinement packets.

[0095] Such a coding allows to distribute the data in different packets depending on their nature, namely :

40 a) essential data
b) non-essential data

45 which are transmitted to the terminal in one or several parallel streams, corresponding to a base stream and different refinement streams.

[0096] Figure 2b thus shows a base packet, which comprise essential data, whereas figures 2c and 2d respectively describe refinement packets and bitplanes refinement packets, which comprise refinement data which can be sent to the terminal through one or several parallel streams.

[0097] Figure 2a presents a typical (non-normative) file structure that may be stored prior to any transmission. The "ZT offsets" shall contain information on the position of the ZT_is in the file, as well as positions of data relative to each bitplane for any ZT_i. As shown in figure 2a, the file structure comprises of:

- a header 20, comprising some information about the type of wavelets used, some parameters of the zero-tree coding, the number of wavelets coefficients per packet, etc.
- 55 - a zone 21 comprising some information about the zero-tree offsets, so as to localise, within the stream, each subarea associated to each zero-tree ;
- scale coefficients 22 ;
- subpackets 23, 24 relating to wavelets coefficients associated to the zero-trees, and to bitplanes.

[0098] The different structures of the received bitstream can be summarised by the tables shown in figures 2b to 2d. The exact structure is given by the syntax specification described hereafter.

[0099] Figure 2b describes the structure of a base packet, comprising

- 5 - a subpacket 30 relating to scale coefficients, allowing to modify the position of the base mesh vertices;
- subpackets 31, 32 relating to wavelets coefficients associated to a zero-tree, containing consecutive bitplanes and at least the first one.

10 [0100] As shown on figure 2c, a refinement packet as a structure similar to that of a base packet, except for the scale coefficients 30.

[0101] Figure 2d presents the structure of a bitplanes refinement packet.

[0102] It is thus possible to distribute the data in packets according to the numerical preciseness required for the rendering of the graphic scene. In other words, the refinement of the mesh also allows for the preciseness on the wavelet coefficients themselves, since the bitplanes refinement packets described in figure 2d allow to send packets 15 which only purpose is to refine the preciseness with which the terminal knows the wavelet coefficients.

[0103] The 3D_Wavelet_Mesh_Object Syntax is as follows :

3D_Wavelet_Mesh_Object

| | | | |
|----|--------------------------------------------------------|-------------|----------|
| 20 | 3D_Wavelet_Mesh_Object () { | No. of bits | Mnemonic |
| 25 | 3D_WMO_start_code | 16 | uimsbf |
| | Base_id | 8 | uimsbf |
| 30 | do { | | |
| | 3D_Wavelet_Mesh_Object_Layer() | | |
| | } while (nextbits_bytealigned() == 3D_WMOL_start_code) | | |
| 35 | } | | |

[0104] The 3D_Wavelet_Mesh_Object_Layer has the following structure :

| | | | |
|----|----------------------------------------|-------------|----------|
| 40 | 3D_Wavelet_Mesh_Object_Layer () { | No. of bits | Mnemonic |
| | 3D_WMOL_start_code | 16 | uimsbf |
| 45 | 3D_WMOL | 1 | |
| | if (!3D_WMOL==0) | | |
| | 3DWaveletMeshObject_Base_Layer() | | |
| | else | | |
| | 3DWaveletMeshObject_Refinement_Layer() | | |
| 50 | } | | |

[0105] We now present the 3D_Wavelet_Mesh_Object_Base_Layer :

55

| | | | |
|----|--------------------------------------------------------|-------------|----------|
| | 3D_Wavelet_MeshObject_Base_Layer() | No. of bits | Mnemonic |
| | do { | | |
| 5 | 3D_WMOBL_start_code | 16 | uimsbf |
| | wmobl_id | 8 | uimsbf |
| | 3D_Wavelet_Mesh_Object_Scale_Coeff() | | |
| | ReadZT(0, wmobl_id) | | |
| 10 | } while (nextbits_bytealigned() == 3D_MOBL_start_code) | | |
| | } | | |

[0106] The following table describes the 3D_Wavelet_Mesh_Object_Scale_Coeff() syntax :

| | | | |
|----|----------------------------------------|-------------|----------|
| 15 | 3D_Wavelet_Mesh_Object_Scale_Coeff() { | No. of bits | Mnemonic |
| | For (i=0 ; i<Nb_sc ; i++) { | | |
| 20 | Sc[i] | NbBpC | bslbf |
| | } | | |
| 25 | } | | |

[0107] The 3D_Wavelet_Mesh_Object_Refinement_Layer syntax is described in the following table :

| | | | |
|----|--------------------------------------------------------|-------------|----------|
| 30 | 3D_Wavelet_MeshObject_Refinement_Layer() | No. of bits | Mnemonic |
| | do { | | |
| 35 | 3D_WMORL_start_code | 16 | uimsbf |
| | wmorl_id | 8 | uimsbf |
| 40 | type | | |
| | ReadZT(type, wmorl_id) | | |
| 45 | } while (nextbits_bytealigned() == 3D_MORL_start_code) | | |
| | } | | |

[0108] The ReadZT() syntax is described hereunder :

50

55

| | | | |
|----|--------------------------------|-------------|----------|
| | ReadZT(A, id) { | No. of bits | Mnemonic |
| 5 | If (A) { | | |
| | FBP | | |
| | LBP | | |
| 10 | } else { | | |
| | FBP=0 | | |
| | LBP=NbBP | | |
| 15 | } | | |
| | For (j=FBP; j<LBP ; i++) { | | |
| 20 | BPLength | 16 | uimsbf |
| | For (k=0 ; k<BPLength ; k++) { | | |
| | Bx[id][k][j] | 1 | |

25

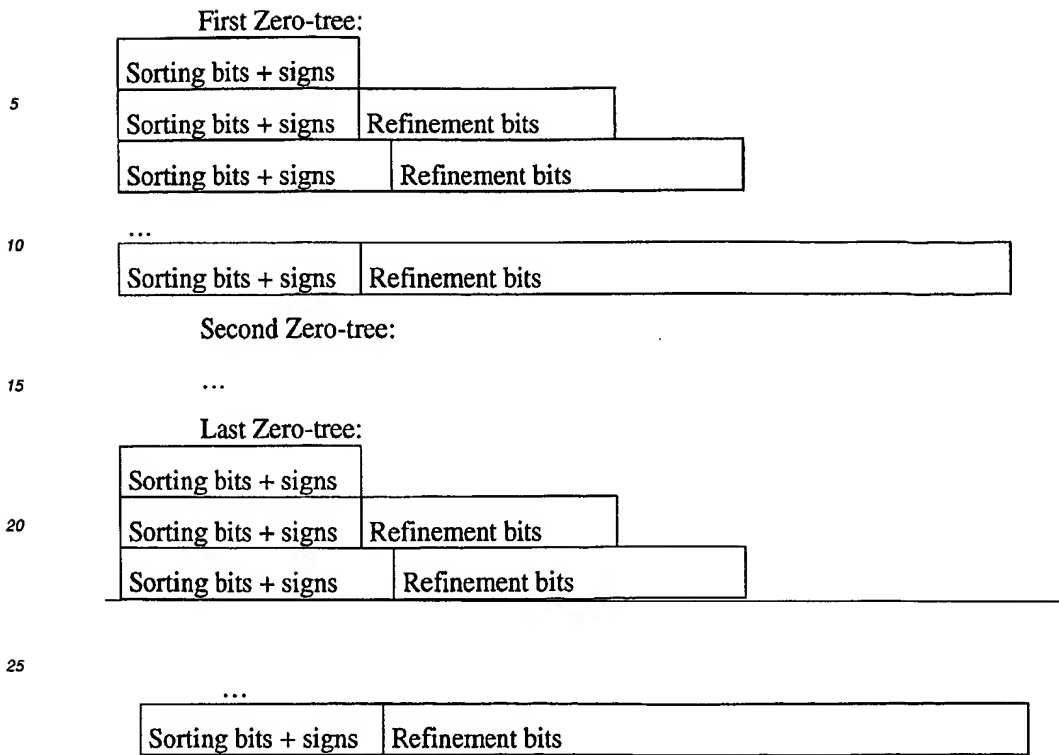
| | | | |
|----|--------------------------------|----|--------|
| | } | | |
| 30 | } | | |
| | For (j=FBP; j<LBP ; i++) { | | |
| 35 | BPLength | 16 | uimsbf |
| | For (k=0 ; k<BPLength ; k++) { | | |
| | By[id][k][j] | 1 | |
| 40 | } | | |
| | } | | |
| | For (j=FBP; j<LBP ; i++) { | | |
| 45 | BPLength | 16 | uimsbf |
| | For (k=0 ; k<BPLength ; k++) { | | |
| | Bz[id][k][j] | 1 | |
| 50 | } | | |
| | } | | |
| 55 | } | | |

A.1.3.5. Semantics

- [0109] 3D_WMO_start_code : This is a unique 16-bit code that is used for synchronisation purpose. The value of this code is always '0000 0000 0010 0000'.
- 5 [0110] Base_id : this 8 bit integer characterises the Wavelet Mesh Object, and is used as a reference for further refinement layers.
- [0111] 3D_WMOL_start_code : 3D_MOL_start_code : This is a unique 16-bit code that is used for synchronisation purposes. The value of this code is always '0000 0000 0011 0000'.
- [0112] 3D_MOL : a boolean with value 0 if and only if the current stream is a base layer.
- 10 [0113] 3D_WMOL_start_code : This is a unique 16-bit code that is used for synchronisation purpose. The value of this code is always '0000 0000 0111 0000'.
- [0114] 3D_WMORL_start_code : This is a unique 16-bit code that is used for synchronisation purpose. The value of this code is always '0000 0000 0010 0000'.
- [0115] wmobl_id : an integer specifying the reference to the object for further refinements.
- 15 [0116] wmorl_id : an integer specifying the reference to the object to refine.
- [0117] BPLength : the length of the current bitplane.
- [0118] Type : a boolean with value 0 if and only if the current bitstream corresponds to type 0 in the bitstream description.
- [0119] FBP : Number of the first bitplane to read.
- 20 [0120] LBP : Number of the last bitplane to read.
- [0121] Bx : A 3-table of bits used for encoding the first component of the wavelet coefficients. The first entry is the number of the zero-tree. The second one is the number of the current bitplane received. The third one is the number of the received bit in this plane.
- [0122] By : A 3-table of bits used for encoding the second component of the wavelet coefficients. The first entry is the number of the zero-tree. The second one is the number of the current bitplane received. The third one is the number of the received bit in this plane.
- 25 [0123] Bz : A 3-table of bits used for encoding the third component of the wavelet coefficients. The first entry is the number of the zero-tree. The second one is the number of the current bitplane received. The third one is the number of the received bit in this plane.
- 30

A.1.3.6 3D Wavelet Mesh Decoding

- [0124] The typical 3D Wavelet Mesh decoding process is figured by the block diagram presented in figure 3.
- [0125] The mesh reconstruction can only begin after transmission and decoding of the base mesh 50.
- 35 [0126] We now give some definitions and notations.
- [0127] Let M_0 denote the base mesh, which is supposed to be entirely transmitted before the wavelet coefficients stream starts being decoded.
- [0128] Let M_j denote the mesh obtained after j iterations of midpoint subdivision of each facet of M_0 .
- [0129] For j non-negative, let Δ_j denote the set of vertices of M_j , and Λ_j the set of vertices of $M_{j+1} \setminus M_j$.
- 40 [0130] Scaling/wavelet coefficients can be indexed by vertices of the reconstructed mesh, and the level of subdivision at which they occur. Hence, we will denote any coefficient c indexed by level j and vertex v as d_v^j .
- [0131] A coefficient d_v^j is said to be offspring of a coefficient d_w^{j-1} if and only if vertex w is a corner vertex of a facet of M_j . The offspring $O(c)$ of a coefficient c is the set of coefficients $\{c_0, c_1, c_2, c_3\}$, as shown in figure 4. The arrows denote the vertices associated to the coefficients, the triangulation figuring the midpoint subdivision of two adjacent facets.
- 45 [0132] The *descendence* $D(c)$ of a coefficient is the transitive closure of the relation "is offspring of".
- [0133] For any 3-vector or 3-vector-valued function c , let c_x, c_y , and c_z denote respectively the first, second and third component of c .
- [0134] While receiving the 3D Wavelet Mesh bitstream 51, the decoder stores in memory the contiguous bits related to a given bitplane in a given Zero-tree. Such a storage can be figured as follows, for each table Bx, By, Bz representing the three components of the wavelet coefficients :
- 50
- 55



[0135] Thus the Zero-tree decoder is completely independent of the bitstream interpretation process. This structure in memory is modified according to the type of bitstream the decoder gets:

Type 0 (as described in figure 2b) :

Base layer. Scaling information, complete header are read, and new Zero-trees are created in memory by the decoder.

Type 1 (as described in figure 2c) :

Refinement layer in which new zero-trees are sent and created in memory by the decoder.

Type 2 (as described in figure 2d) :

Bitplanes refinements of previously sent zero-trees.
 This structure corresponds to the entries of table B : $B[i][j][k]$ is stored in the i-th zero-tree space, at line j and column k.
 In the Zero-tree decoder 52, the algorithm described in the previously mentioned article by A. Said and W. A. Pearlman is applied as is with previous definition of offspring and descendants, on each table Bx, By and Bz for recovering respectively a list Lx, Ly and Lz of projected wavelet coefficients.

A.1.3.7 Synthesis filters and mesh reconstruction

[0136] We describe here the standard, non-algorithmic mesh reconstruction process using the above decoded/stored wavelet coefficients. The scope of adaptability will thus be restricted to the transmission : no normative directives will constrain the way the client selects the data contained in his cache for partially reconstructing the mesh, according to frame-rate, LoD management etc. Such a reconstruction implements the synthesis filters 53 in order to deliver a reconstructed 3D Wavelet mesh 54.

[0137] The behaviour of the terminal designed to render the graphic scene could be described in other words as

follows. The terminal first receives a scene graph comprising a node associated to an object which mesh is coded according to a wavelet encoding technique. Such a node also comprises means to configure a wavelet decoder called "DecoderConfig".

[0138] The node comprises the definition of a base mesh, some references to compressed wavelet coefficients streams, some parameters allowing to adjust the behaviour of the node, as well as its modification within the scene graph, and optionally a description of a back channel, as mentioned previously in the document.

[0139] The terminal instantiates a decoder thanks to "DecoderConfig" and opens one or several data channels associated to the decoder. In case there are several channels, one of them is considered as the base channel, while the others are refinement channels, as already mentioned previously).

[0140] The data received by the terminal through this or these channel(s) are decoded, in order to get the wavelet coefficients, which are stored in a cache associated to the terminal (for example, the stocking/destocking takes place in a buffer memory).

[0141] According to the position of the user in the scene, the rendering system of the terminal decides to take into account a given number of wavelet coefficients and to apply them to the base mesh, which allows to obtain a mesh which is adapted to the visualisation conditions. Such a mesh is then displayed on the screen of the terminal.

[0142] The topology (facets configuration) of the fully reconstructed mesh is semi-regular : it corresponds to the topology of M_j .

[0143] The geometry (vertices position in space) of the mesh can be recovered as follows :

For j non-negative, let T^j and B^j be matrices as specified in Annex 4.

Let $(c^0)_i$ be the decoded scale coefficients.

Then the positions c_j^i of respective vertices v_i can be recovered by iterating the following relations :

$$c_x^{j+1} = T^j (c_x^j f_x(d^j)^t)^t$$

$$c_y^{j+1} = T^j (c_y^j f_y(d^j)^t)^t$$

$$c_z^{j+1} = T^j (c_z^j f_z(d^j)^t)^t$$

over $j=jmax, ..., 0$, Function f , which is a basis change transformation from local coordinates to global coordinates is explained in Annex 4.

[0144] Of course, in practice, one can exploit the sparsity of matrices T^j , and the fact that only a portion of the (ϕ) (and only the corresponding non-homogeneous semi-regular topology) need to be used in the reconstructed mesh.

A.1.4 BackChannel : the monitoring channel.

[0145] The BackChannel is the stream conveying the information necessary to monitor the state of the terminal (cache state, viewpoint).

A.1.4.1 Functionality.

[0146] The BackChannel is an upstream elementary stream which allows the client to request for specific chunks of the wavelet coefficients so that only relevant parts of the mesh/uvMap are refined. It also allows the server to know the client state and send in advance chunks that may be useful.

A.1.4.2 ES Configuration. DecoderSpecificInfo

[0147] The configuration is defined as follows.

A.1.4.2.1 Stream Configuration : Syntax

[0148]

```

5      class WmeshBC_DecoderConfig extends DecoderSpecificInfo : bit(8)
        tag=DecSpecificInfoTag {
10         bit(5) viewpointNbBits
        }

```

A.1.4.2.2 Semantics

15 [0149] ViewpointNbBits is the number of bits used to quantize the viewpoint direction.

A.1.4.3 BackChannelFrame.

20 A.1.4.3.1 Syntax

[0150]

```

25      class BackChannelCommandFrame() {
        Position position;
        Viewpoint viewpoint;
30         bit(1) reserved

        bit(1) hasMeshCacheState
        if (hasMeshCacheState)
        {
40             CacheState meshCacheState;
        }
        else
45         {
            CacheDelta meshCacheDelta;
50         }
    }

```

A.1.4.3.2 Semantics.

55 [0151] This is the payload of an access unit (AU), as defined in annex 1.

[0152] First the position and the viewpoint direction are sent. The position is described in the local coordinate system of the WaveMesh node.

A.1.4.4 PositionA.1.4.4.1 Syntax

5 [0153]

```

class Position () {
10     EfficientFloat x
        EfficientFloat y
        EfficientFloat z
15     }

```

A.1.4.4.1 Semantics

20 [0154] The Position syntax describes the position of the user with respect to the local coordinate system of the user.

A.1.4.5 ViewpointA.1.4.5.1 Syntax

25

[0155]

```

class Viewpoint () {
30     int(1) direction
        int(2) orientation
        int(viewpointNbBits) vq[0]
35
        int(viewpointNbBits) vq[1]
40     }

```

A.1.4.5.2 Semantics.

45 [0156] The Viewpoint syntax describes the viewing direction of the user with respect to the local coordinate system of the user. It uses the quantization scheme defined in 9.3.7.25 in ISO/IEC 14496-1 with a fixed number of bits defined in the decoder specific info.

50

55

A.1.4.6 CacheStateA.1.4.6.1 Syntax

5 [0157]

```

class CacheState () {
10     bit(2) reserved
        int i=0;
        for (i=0;i<numFaces;i++)
15     {
            int(8) numWaveletCoefficients
            int(5) numBitPlane
20     }
    }

```

25 A.1.4.6.2 Semantics.

[0158] This element specifies for all the faces of the base mesh the number of coefficients as well as the number of bitplanes received concerning those coefficients.

30 A.1.4.7 DeltaCacheA.1.4.7.1 Syntax

[0159]

35

```

class DeltaCache () {
        bit(2) reserved
40     int i=0;
        int(5) numFaceBits;
        int nbBits=numFaceBits;
45

```

50

55

```

    for (faceNb =0;i<numFaces;i++)
5      {
        if (baseMeshFace[faceNb].cacheStateHasChanged())
        {
10          int (nbBits) faceNb;
          int(8) numWaveletCoefficients';
          int(5) numBitPlane;
15        }
      }
20  }

```

A.1.4.7.2 Semantics.

[0160] This element specifies for all the faces of the base mesh the number of coefficients as well as the number of bitplanes received concerning those coefficients.

ANNEX 1

DEFINITIONS :

30

1. Access Unit (AU)

[0161] An individually accessible portion of data within an *elementary stream*. An access unit is the smallest data entity to which timing information can be attributed.

35

2. Audio-visual Object

[0162] A representation of a natural or synthetic object that has an audio and/or visual manifestation. The representation corresponds to a node or a group of nodes in the BIFS scene description. Each audio-visual object is associated with zero or more *elementary streams* using one or more *object descriptors*.

40

3. Audio-visual Scene (AV Scene)

[0163] A set of audio-visual objects together with scene description information that defines their spatial and temporal attributes including behaviours resulting from object and user interactions.

45

4. Binary Format for Scene (BIFS)

[0164] A coded representation of a parametric scene description format.

50

5. Terminal

[0165] A system that sends, or receives and presents the coded representation of an interactive audio-visual scene as defined by ISO/IEC 14496-1. It can be a standalone system, or part of an application system complying with ISO/IEC 14496.

55

ANNEX 2

Abbreviations and Symbols

5 [0166]

| | |
|-----------|--------------------------------------------------|
| AU | Access Unit |
| AV | Audio-visual |
| BIFS | Binary Format for Scene |
| 10 CM | Composition Memory |
| CTS | Composition Time Stamp |
| CU | Composition Unit |
| DAI | DMIF Application Interface (see ISO/IEC 14496-6) |
| DB | Decoding Buffer |
| 15 DTS | Decoding Time Stamp |
| ES | Elementary Stream |
| ESI | Elementary Stream Interface |
| ESID | Elementary Stream Identifier |
| FAP | Facial Animation Parameters |
| 20 FAPU | FAP Units |
| NCT | Node Coding Tables |
| NDT | Node Data Type |
| OCI | Object Content Information |
| OCR | Object Clock Reference |
| 25 OD | Object Descriptor |
| ODID | Object Descriptor Identifier |
| OTB | Object Time Base |
| PLL | Phase Locked Loop |
| QoS | Quality of Service |
| 30 SDM | Systems Decoder Model |
| SL | Synchronization Layer |
| SL-Packet | Synchronization Layer Packet |
| SPS | SL-Packetized Stream |
| STB | System Time Base |
| 35 TTS | Text-To-Speech |
| URL | Universal Resource Locator |
| VOP | Video Object Plane |
| VRML | Virtual Reality Modeling Language |

40 **ANNEX 3**

WaveMesh

45 [0167]

50

55

| WaveMesh | SFWorldNode SF3DNode | | | | | 0110000 01001 | | |
|-------------|-------------------------|--------|-------|--------|--------|------------------|---|---|
| Field name | Field type | DEF Id | IN Id | OUT Id | DYN Id | [m, M] | Q | A |
| backChannel | MFUrl | 000 | | | | | | |
| baseMesh | SFNode | 001 | | | | | | |
| meshStream | MFUrl | 010 | | | | | | |
| texture | SFUrl | 100 | 00 | 00 | | | | |
| frequency | SFFloat | 101 | 01 | 01 | | [0,100] | | |

(continued)

| | | | | | | | | |
|------------|-------------|--------|-------|--------|--------|---------|---|---|
| WaveMesh | SFWorldNode | | | | | 0110000 | | |
| | SF3DNode | | | | | 01001 | | |
| Field name | Field type | DEF id | IN id | OUT id | DYN id | [m, M] | Q | A |
| quality | SFInt32 | 110 | 10 | 10 | | [0,2] | | |

10 **ANNEX 4****Synthesis filters and basis changes**

15 **[0168]** For a purpose of clarity, vertices will be identified to their indices. Let $d_j(i,k)$ denote the function returning the shortest edge-path on M_j between vertices i and k .

Let $\text{level}(i)$ denote the function returning the smallest j so that vertex i belongs to M_j .

Let S_j denote the mesh having the same connectivity as M_j but whose vertices have c_j^i as coordinates : vertex i of S_j has c_j^i as coordinates.

20 We will say that a triangle (abc) on an orientable surface is clockwise oriented if it is clockwise oriented with respect to the surface orientation.

Filters for Loop wavelets are defined as:

$T_l = (P_l \ Q_l)$

with $P_l = (p_{ik})$, $Q_l = (q_{ik})$

25 where $p_{ki} = 10$ if $i=k$

6 if $d_{j+1}(i,k)=1$

2 if $d_{j+1}(i,k)=2$ and $\text{level}(k)=\text{level}(i)-1$

1 if $d_{j+1}(i,k)=2$ and $\text{level}(k)=\text{level}(i)$

30 0 otherwise

and $q_{ki} = 10$ if $i=k$

-6 if $d_{j+1}(i,k)=1$ and $\text{level}(k)=j-1$

35 -6 if $d_{j+1}(i,k)=1$ and there exists m such that

$\text{level}(m)=j-1$

$d_{j+1}(m,i)=1$

(kmi) is anticlockwise oriented.

6 if $d_{j+1}(i,k)=1$ and there exists m such that

40 $\text{level}(m)=j-1$

$d_{j+1}(m,i)=1$

(kmi) is clockwise oriented.

1 if $d_{j+1}(i,k)=2$ and there does not exist two different m and n such that

$d_{j+1}(n,i)=1$ and $d_{j+1}(m,i)=1$

and $d_{j+1}(m,k)=1$

45 and $d_{j+1}(n,k)=1$.

-2 if $d_{j+1}(i,k)=2$ and there exist m and n such that

$d_{j+1}(m,i)=1$ and

$d_{j+1}(n,i)=1$ and $q_{ni}=-6$ and $q_{mi}=-6$

50 -2 for any other j such that $d_{j+1}(i,k)=2$

0 otherwise.

For lifted midpoint subdivision with one vanishing moment:

For any vertex v of M_{j+1} , let a and b denote the vertices of the edge v of which v is the midpoint. Then

55 $p_{ki} = 1$ if $i=k$

1/2 if $d_{j+1}(i,k)=1$

0 otherwise

$q_{ki} = A_a/2 + A_b/2 + 1$ if $i=k$

A_a if $k=a$

A_b if $k=b$

5 $A_a/2$ if $d_{j+1}(k,a)=1$ and $k \neq i$

$A_b/2$ if $d_{j+1}(k,b)=1$ and $k \neq i$

0 otherwise

with $A_x = -(3/4) * (\sin(\pi/3)) / (n_x * \sin(2 * \pi/n_x))$, n_x being the valence of x .

Basis change:

10 For $j > 0$, $f(d_j)$ is defined as follows :

Let $A=[b,c]$ be the edge of M_j of which vertex i in M_{j+1} is the midpoint. Let

N, U, V be the vectors defined as follows:

15 [0169] If there exist two different facets in S_j containing A , then N is the normalized average of the normals to those two facets. Otherwise, it is the normal to the only facet of S_j having A as an edge.

$$U := \vec{bc} / \|\vec{bc}\|$$

20

$$V := NxU$$

Then $f(d_j) : (NUV)D_j$

25

Claims

1. Method for rendering a graphic scene on a terminal, said graphic scene implementing a hierarchical tree comprising at least one node associated to an object of said scene,
30 wherein at least one of said objects is defined by a wavelet-coded mesh, as a base mesh and wavelet coefficients corresponding to local modifications of said base mesh.
2. Method according to claim 1, wherein said graphic scene is of the MPEG4 type.
- 35 3. Method according to any of claims 1 and 2, wherein, for each of said objects defined by a wavelet-coded mesh, said corresponding node is associated, within a node interface of said hierarchical tree, to :
 - a field relating to said base mesh, called BaseMesh ;
 - a field relating to at least one rendering stream of said object, comprising said wavelets coefficients, called
40 MeshStream,

and wherein said BaseMesh and MeshStream fields can be transmitted and/or processed independently.
- 45 4. Method according to any of claims 1 to 3, wherein said terminal has a back channel, allowing to transmit at least one piece of information relating to said terminal and/or to an user of said terminal to a graphic scene server.
5. Method according to claim 4, wherein, for each of said objects defined by a wavelet-coded mesh, said corresponding node interface comprises a field allowing to indicate said back channel.
- 50 6. Method according to any of claims 4 and 5, wherein said piece of information belongs to the group comprising
 - a piece of information relating to the position of said user in said graphic scene, called position information ;
 - a piece of information relating to a direction of observation of said user, called direction information ;
 - a piece of information relating to the state of a cache of said terminal.
- 55 7. Method according to any of claims 4 to 6, wherein said node interface also implements at least one of the fields belonging to the group comprising

- a field relating to the texture of said object;
 - a field indicating a frequency at which said terminal transmits at least one state information to a server ;
 - a field relating to a field of view of said user, called FOV ;
 - a field indicating a quality of rendering of said object, called Quality.
- 5
8. Method according to any of claims 3 to 7, wherein said node interface also comprises a field, called FocusPoint, allowing to indicate at least one area of said object for which an user wishes to get a quality of rendering greater than a predetermined threshold.
- 10
9. Method according to any of claims 7 and 8, wherein said Quality field pilots a relevance threshold for said wavelets coefficients, the wavelets coefficients lower than said relevance threshold not being applied to said mesh.
- 15
10. Method according to any of claims 3 to 9, wherein at least one of said objects may be animated by modifying in time, within said BaseMesh field, coordinates of at least one vertex of said base mesh, the wavelets coefficients of said MeshStream field remaining unchanged.
11. Method according to any of claims 3 to 10, wherein a step of editing said mesh allows to modify an existing object by acting on at least one of the elements belonging to the group comprising
- 20
- said base mesh;
 - said wavelets coefficients ;
 - a current mesh associated to said existing object within said rendered scene.
- 25
12. Method according to any of claims 3 to 11, wherein said rendering stream of said MeshStream field is organised into packets of at least one wavelets coefficient.
13. Method according to claim 12, wherein said packets of said rendering stream are of at least two types of packets, among the types belonging to the group comprising:
- 30
- base packets, comprising data which are essential to the rendering of said mesh defining said object;
 - refinement packets ;
 - bitplanes refinement packets,
- said refinement packets and said bitplanes refinement packets comprising data for refining said mesh.
- 35
14. Method according to claim 13, wherein said refinement packets comprise at least one wavelets coefficient allowing to refine the rendering of at least one area of said object.
15. Method according to any of claims 13 and 14, wherein said bitplanes refinement packets allow to refine the preciseness of at least one wavelets coefficient.
- 40
16. Method according to any of claims 12 to 15, wherein said packets are transmitted from a server to at least one terminal within a single rendering stream.
- 45
17. Method according to any of claims 12 to 15, wherein said packets are transmitted from a server to at least one terminal within at least two parallel data streams, corresponding respectively to a base stream, comprising at least one base packet, and to at least one refinement stream, comprising at least one refinement packet and/or at least one bitplanes refinement packet.
- 50
18. Method according to any of claims 6 to 17, wherein said information relating to the state of a cache of said terminal indicates at least:
- a number of wavelets coefficients associated to each packet stored within said cache;
 - a number of bitplanes associated to each packet stored within said cache.
- 55
19. Method according to any of claims 12 to 18, wherein said wavelets coefficients are organised into at least two separate subsets, extracted from a global data stream allowing to rebuild said object, each of said subsets corresponding to areas of said object, and undergoing independent coding, localisation data, which allow to locate said

separate subsets within said global data stream, being inserted within said global data stream, so as to allow to distribute said wavelets coefficients among said packets, in view of a selective transmission of at least one of said subsets towards said terminal.

- 5 **20.** Method according to claim 19, wherein said subsets undergo a coding of "zero-tree" type.
- 21.** Method according to any of claims 13 to 20, wherein each type of packet is structured into subpacket(s) belonging to the group comprising
- 10 - for a base packet:
- a subpacket relating to scale coefficients, allowing to modify the position of at least one vertex of said base mesh;
- at least one subpacket relating to wavelets coefficients associated to a zero-tree, containing consecutive bitplanes and at least the first one ;
- 15 - for a refinement packet:
- at least one subpacket relating to wavelets coefficients associated to a zero-tree, containing consecutive bitplanes and at least the first one ;
- 20 - for a bitplanes refinement packet:
- at least one subpacket relating to wavelets coefficients associated to a zero-tree, containing consecutive bitplanes.
- 25 **22.** Method according to any of claims 20 and 21, wherein said global data stream comprises a header, comprising at least some of said localisation data, and a wavelets coefficients area, comprising a subarea identified by said localisation data for each of said subsets.
- 30 **23.** Method according to claim 22, wherein said header also comprises at least some of the information belonging to the group comprising :
- the number of faces of said base mesh ;
- 35 - the type of wavelets ;
- information relating to said object;
- information relating to the coding of said localisation data.
- 40 **24.** Method according to any of claims 1 to 23, wherein a wavelets coded texture is also associated to at least one of said objects defined by a wavelets coded mesh.
- 25.** Terminal **characterised in that** it renders at least one graphic scene according to the method of any of claims 1 to 24.
- 45 **26.** Server of graphic scenes **characterised in that** it transmits to at least one client terminal a graphic scene designed to be rendered according to the method of any of claims 1 to 24.
- 27.** Data storage medium **characterised in that** it comprises at least one graphic scene designed to be rendered by at least one client terminal according to the method of any of claims 1 to 24.
- 50 **28.** Method for coding a graphic scene implementing a hierarchical tree comprising at least one node associated to an object,
- wherein at least one of said objects is defined by a wavelets coded mesh, as a base mesh and wavelets coefficients corresponding to local modifications of said base mesh.
- 55 **29.** Method according to claim 28, wherein said graphic scene is designed to be rendered on at least one client terminal according to the method of any of claims 1 to 24.

30. Graphic scene implementing a hierarchical tree comprising at least one node associated to an object, wherein at least one of said objects is defined by a wavelets coded mesh, as a base mesh and wavelets coefficients corresponding to local modifications of said base mesh.
- 5 31. Node of a hierarchical tree representative of a graphic scene, **characterised in that** it is associated to an object defined by a wavelets coded mesh, as a base mesh and wavelets coefficients corresponding to local modifications of said base mesh.
- 10 32. Signal to be transmitted to at least one terminal for rendering a graphic scene on said terminal, said signal comprising data representative of said graphic scene implementing a hierarchical tree comprising at least one node associated to an object of said scene, wherein said signal comprises at least one object defined by a wavelets coded mesh, as a base mesh and wavelets coefficients corresponding to local modifications of said base mesh.
- 15 33. Signal according to claim 32, wherein said signal is of the MPEG4 type.
34. Signal according to any of claims 32 and 33, **characterised in that**, for each of said objects defined by a wavelet-coded mesh, it comprises
- 20 - a field relating to said base mesh, called BaseMesh ;
- a field relating to a rendering stream of said object, comprising said wavelets coefficients, called MeshStream.
35. Signal according to any of claims 32 to 34, **characterised in that** it comprises a field allowing to indicate a back channel of said terminal, said back channel allowing to transmit at least one piece of information relating to said terminal and/or to an user of said terminal to a graphic scene server.
- 25 36. Signal according to claim 35, wherein said piece of information belongs to the group comprising
- 30 - a piece of information relating to the position of said user in said graphic scene, called position information ;
- a piece of information relating to a direction of observation of said user, called direction information ;
- a piece of information relating to the state of a cache of said terminal.
37. Signal according to any of claims 35 and 36, **characterised in that** it also comprises at least one of the fields belonging to the group comprising
- 35 - a field relating to the texture of said object ;
- a field indicating a frequency at which said terminal transmits at least one state information to a server ;
- a field relating to a field of view of said user, called FOV ;
- a field indicating a quality of rendering of said object, called Quality.
- 40 38. Signal according to any of claims 34 to 37, **characterised in that** it also comprises a field, called FocusPoint, allowing to indicate at least one area of said object for which an user wishes to get a quality of rendering greater than a predetermined threshold.
- 45 39. Signal according to any of claims 37 and 38, wherein said Quality field pilots a relevance threshold for said wavelets coefficients, the wavelets coefficients lower than said relevance threshold not being applied to said mesh.
40. Signal according to any of claims 34 to 39, wherein said rendering stream of said MeshStream field is organised into packets of at least one wavelets coefficient.
- 50 41. Signal according to claim 40, wherein said packets of said rendering stream are of at least two types of packets, among the types belonging to the group comprising
- 55 - base packets, comprising data which are essential to the rendering of said mesh defining said object ;
- refinement packets ;
- bitplanes refinement packets,
said refinement packets and said bitplanes refinement packets comprising data for refining said mesh.

42. Signal according to claim 41, wherein said refinement packets comprise at least one wavelets coefficient allowing to refine the rendering of at least one area of said object.
- 5 43. Signal according to any of claims 41 and 42, wherein said bitplanes refinement packets allow to refine the preciseness of at least one wavelets coefficient.
44. Signal according to any of claims 40 to 43, wherein said packets are transmitted from a server to at least one terminal within a single rendering stream.
- 10 45. Signal according to any of claims 40 to 43, wherein said packets are transmitted from a server to at least one terminal within at least two parallel data streams, corresponding respectively to a base stream, comprising at least one base packet, and to at least one refinement stream, comprising at least one refinement packet and/or at least one bitplanes refinement packet.
- 15 46. Signal according to any of claims 36 to 45, wherein said information relating to the state of a cache of said terminal indicates at least:
- a number of wavelets coefficients associated to each packet stored within said cache;
 - a number of bitplanes associated to each packet stored within said cache.
- 20 47. Signal according to any of claims 40 to 46, wherein said wavelets coefficients are organised into at least two separate subsets, extracted from a global data stream allowing to rebuild said object, each of said subsets corresponding to areas of said object, and undergoing independent coding, localisation data, which allow to locate said separate subsets within said global data stream, being inserted within said global data stream, so as to allow to
- 25 distribute said wavelets coefficients among said packets, in view of a selective transmission of at least one of said subsets towards said terminal.
48. Signal according to claim 47, wherein said subsets undergo a coding of "zero-tree" type.
- 30 49. Signal according to any of claims 41 to 48, wherein each type of packet is structured into subpacket(s) belonging to the group comprising
- for a base packet:
 - 35 - a subpacket relating to scale coefficients, allowing to modify the position of at least one vertex of said base mesh ;
 - at least one subpacket relating to wavelets coefficients associated to a zero-tree, containing consecutive bitplanes and at least the first one ;
 - 40 - for a refinement packet:
 - at least one subpacket relating to wavelets coefficients associated to a zero-tree, containing consecutive bitplanes and at least the first one ;
 - 45 - for a bitplanes refinement packet:
 - at least one subpacket relating to wavelets coefficients associated to a zero-tree, containing consecutive bitplanes.
- 50 50. Signal according to any of claims 47 to 49, wherein said global data stream comprises a header, comprising at least some of said localisation data, and a wavelets coefficients area, comprising a subarea identified by said localisation data for each of said subsets.
- 55 51. Signal according to claim 50, wherein said header also comprises at least some of the information belonging to the group comprising
- the number of faces of said base mesh ;
 - the type of wavelets ;

EP 1 316 922 A1

- information relating to said object;
- information relating to the coding of said localisation data.

5 52. Signal according to any of claims 32 to 51, wherein it comprises a wavelets coded texture associated to at least one of said objects defined by a wavelets coded mesh.

53. Device **characterised in that** it comprises means for emitting a signal according to any of claims 32 to 52.

10 54. Device **characterised in that** it comprises means for receiving a signal according to any of claims 32 to 52.

15

20

25

30

35

40

45

50

55

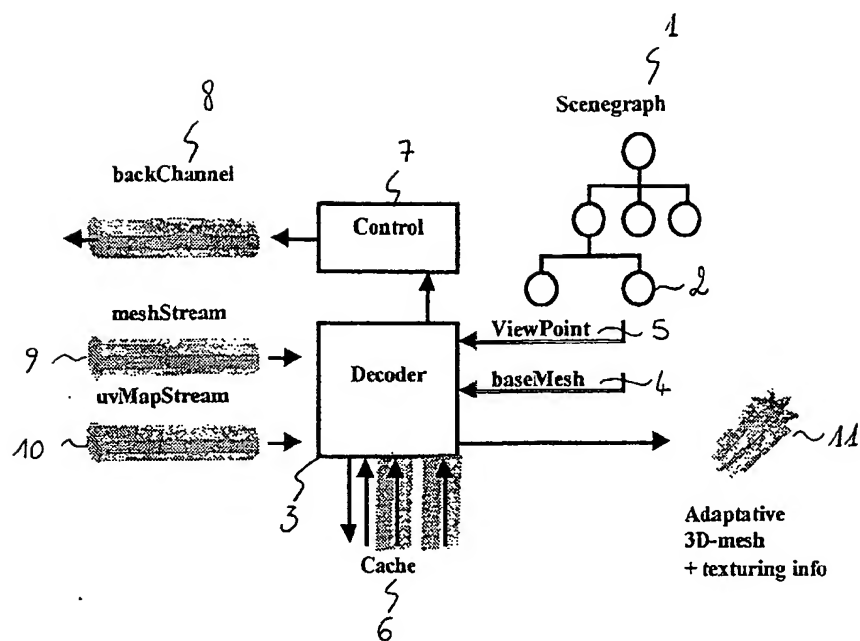


FIGURE 1

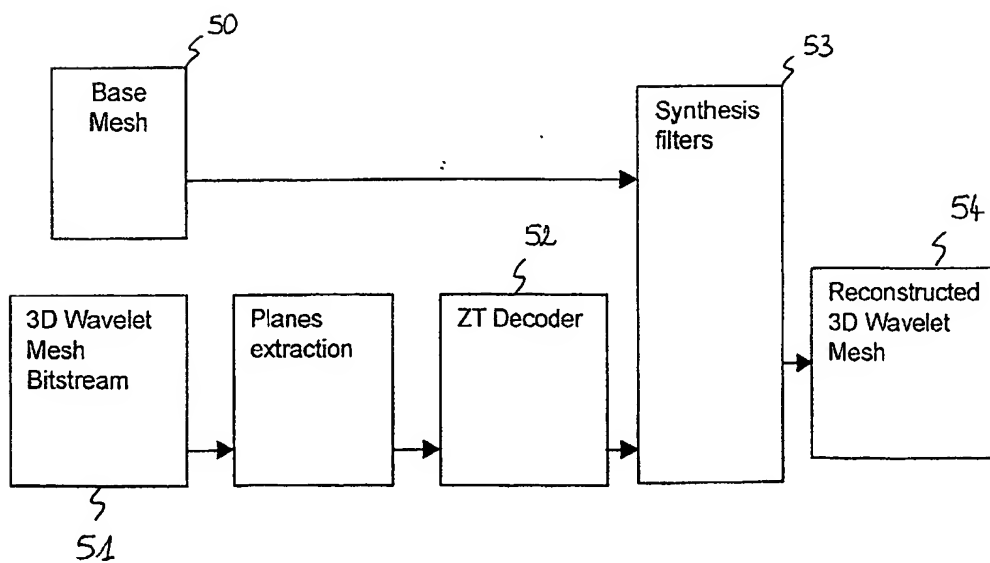


FIGURE 3

BEST AVAILABLE COPY

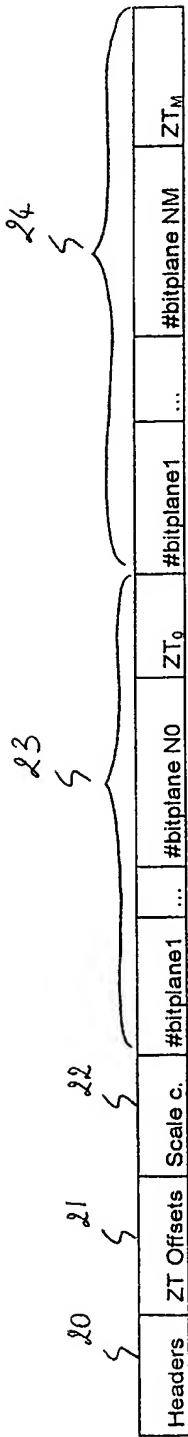


FIGURE 2A

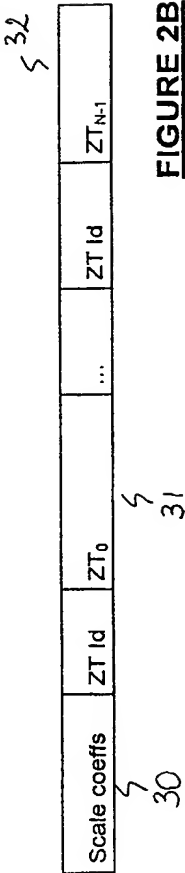


FIGURE 2B



FIGURE 2C

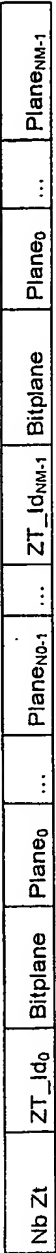


FIGURE 2D

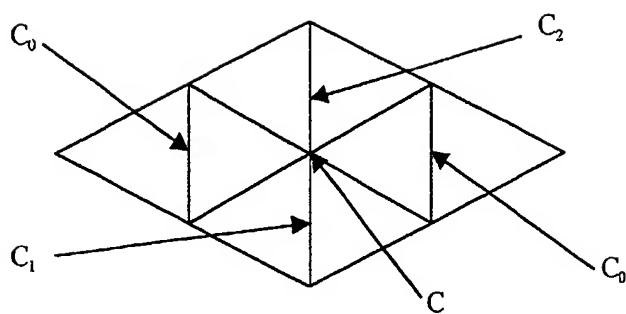


FIGURE 4



European Patent
Office

EUROPEAN SEARCH REPORT

Application Number
EP 01 46 0075

| DOCUMENTS CONSIDERED TO BE RELEVANT | | | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------|
| Category | Citation of document with indication, where appropriate, of relevant passages | Relevant to claim | CLASSIFICATION OF THE APPLICATION (Int.Cl.7) |
| X | MIN SHENG ZHONG ET AL: "JPEG2000-based scalable reconstruction of image local regions" PROCEEDINGS OF 2001 INTERNATIONAL SYMPOSIUM ON INTELLIGENT MULTIMEDIA, VIDEO AND SPEECH PROCESSING. ISIMP 2001 (IEEE CAT. NO.01EX489), HON, pages 174-177, XP002211321 2001, Piscataway, NJ, USA, IEEE, USA ISBN: 962-85766-2-3 | 1-3,12, 16,19, 20,22, 23, 25-34, 40,44, 47,48, 50,51, 53,54 | G06T17/00 |
| Y | | 4,7,24, 35,37,52 | |
| A | * abstract * | 5,6, 8-11, 13-15, 17,18, 21,36, 38,39, 41-43, 45,46,49 | |
| | * sections II, III.A 1)-3), III.B --- | | TECHNICAL FIELDS SEARCHED (Int.Cl.7) |
| D,X | KHODAKOVSKY A ET AL: "PROGRESSIVE GEOMETRY COMPRESSION" COMPUTER GRAPHICS. SIGGRAPH 2000 CONFERENCE PROCEEDINGS. NEW ORLEANS, LA, JULY 23 - 28, 2000, COMPUTER GRAPHICS PROCEEDINGS. SIGGRAPH, NEW YORK, NY: ACM, US, 23 July 2000 (2000-07-23), pages 271-278, XP001003565 ISBN: 1-58113-208-5 | 1,25-32, 53,54 | G06T |
| Y | * sections 1.1, 3.2 --- -/-- | 2,33 | |
| The present search report has been drawn up for all claims | | | |
| Place of search BERLIN | | Date of completion of the search 28 August 2002 | Examiner Kröner, S |
| CATEGORY OF CITED DOCUMENTS X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document | | T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons & : member of the same patent family, corresponding document | |

EPO FORM 1503 03.92 (P04001)



European Patent
Office

EUROPEAN SEARCH REPORT

Application Number
EP 01 46 0075

| DOCUMENTS CONSIDERED TO BE RELEVANT | | | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------|
| Category | Citation of document with indication, where appropriate, of relevant passages | Relevant to claim | CLASSIFICATION OF THE APPLICATION (Int.Cl.7) |
| D,X | LOUNSBERY M ET AL: "MULTIRESOLUTION ANALYSIS FOR SURFACES OF ARBITRARY TOPOLOGICAL TYPE" ACM TRANSACTIONS ON GRAPHICS, ASSOCIATION FOR COMPUTING MACHINERY, NEW YORK, US, vol. 16, no. 1, 1997, pages 34-73, XP000686202 ISSN: 0730-0301 * abstract; figure 1 * * sections 1, 2 * | 1,25-32, 53,54 | TECHNICAL FIELDS SEARCHED (Int.Cl.7) |
| Y | JORDAN F ET AL: "Streaming of photo-realistic texture mapped on 3D surface" PROCEEDINGS OF THE 1997 INTERNATIONAL CONFERENCE ON IMAGE PROCESSING, SANTA BARBARA, CA, USA 26-29 OCT. 1997, LOS ALAMITOS, CA, USA, IEEE COMPUT. SOC, US, 26 October 1997 (1997-10-26), pages 390-393, XP010253985 ISBN: 0-8186-8183-7 * abstract * * sections 1, 3.3, 3.4, 4.4, 5.1 * | 4,7,24, 35,37,52 | |
| Y | BEREKOVIC M ET AL: "A flexible processor architecture for MPEG-4 image compositing" PROCEEDINGS OF THE 1998 IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING, 1998, SEATTLE, WA, USA 12-15 MAY 1998, NEW YORK, NY, USA, IEEE, US, 12 May 1998 (1998-05-12), pages 3153-3156, XP010279508 ISBN: 0-7803-4428-6 * sections 1, 2 * * abstract * | 2,33 | |
| A | --- | 4,24,35, 52 | |
| The present search report has been drawn up for all claims | | | |
| Place of search BERLIN | | Date of completion of the search 28 August 2002 | Examiner Kröner, S |
| CATEGORY OF CITED DOCUMENTS X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document | | T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons & : member of the same patent family, corresponding document | |

EPO FORM 1503 03.82 (P04C01)



European Patent
Office

EUROPEAN SEARCH REPORT

Application Number
EP 01 46 0075

| DOCUMENTS CONSIDERED TO BE RELEVANT | | | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------|
| Category | Citation of document with indication, where appropriate, of relevant passages | Relevant to claim | CLASSIFICATION OF THE APPLICATION (Int.Cl.7) |
| A | <p>PURI A ET AL: "Implicit arbitrary shape visual objects via MPEG-4 scene description"</p> <p>PROCEEDINGS OF THE 1999 INTERNATIONAL CONFERENCE ON IMAGE PROCESSING, 1999. ICIP 99. KOBE, JAPAN 24-28 OCT. 1999, PISCATAWAY, NJ, USA, IEEE, US, 24 October 1999 (1999-10-24), pages 258-262, XP010369199</p> <p>ISBN: 0-7803-5467-2</p> <p>* sections 1, 3.3 *</p> <p>-----</p> | 2,33 | |
| | | | TECHNICAL FIELDS SEARCHED (Int.Cl.7) |
| | | | |
| The present search report has been drawn up for all claims | | | |
| Place of search BERLIN | | Date of completion of the search 28 August 2002 | Examiner Kröner, S |
| CATEGORY OF CITED DOCUMENTS | | <p>T : theory or principle underlying the invention</p> <p>E : earlier patent document, but published on, or after the filing date</p> <p>D : document cited in the application</p> <p>L : document cited for other reasons</p> <p>.....</p> <p>& : member of the same patent family, corresponding document</p> | |
| <p>X : particularly relevant if taken alone</p> <p>Y : particularly relevant if combined with another document of the same category</p> <p>A : technological background</p> <p>O : non-written disclosure</p> <p>P : intermediate document</p> | | | |

EPO FORM 1503 03.82 (P04C01)